

# Determining Relevant Relations for Datalog Queries under Access Limitations is Undecidable

[Extended Abstract]

Davide Martinenghi  
Politecnico di Milano  
Piazza Leonardo 32  
20132 Milano, Italy  
davide.martinenghi@polimi.it

## ABSTRACT

Access limitations are restrictions in the way in which the tuples of a relation can be accessed. Under access limitations, query answering becomes more complex than in the traditional case, with no guarantee that the answer tuples that can be extracted (aka *maximal answer*) are all those that would be found without access limitations (aka *complete answer*). The field of query answering under access limitations has been broadly investigated in the past. Attention has been devoted to the problem of determining relations that are *relevant* for a query, i.e., those (possibly off-query) relations that might need to be accessed in order to find all tuples in the maximal answer. In this short paper, we show that relevance is undecidable for Datalog queries.

## 1. RELEVANCE

The problem of querying data sources that have limited capabilities and can thus only be accessed by providing values for certain fields according to given patterns has raised a great deal of interest in the past few years [17, 18, 19, 26, 15, 16, 20, 22, 23, 21, 10, 13, 12, 25, 11, 28, 9, 27].

An access pattern is a constraint indicating which attributes of a relation schema are used as input and which ones are used as output.

In this respect, access patterns may suitably characterize several relevant contexts, such as Web forms, legacy data, Web services, and the so-called Deep Web [6, 7, 24]. Query processing under access patterns requires specialized techniques. Among these, static optimization, including query containment, has been studied for several forms of conjunctive queries and unions thereof [4, 5, 2, 8, 21, 1]. More general cases are covered in the context of dynamic optimization [3], where results are available for schemata with functional dependencies and simple full-width inclusion dependencies. The latter kind of dependencies, albeit simple,

can be used to state equivalence, and thus captures the notion of relations with multiple access patterns.

In the context of access limitations, it is important to distinguish between the maximal answer and the complete answer to a query. The *maximal answer* is the largest set of query answers that can be computed from the relations in the schema over which the query is posed, while complying with the access limitations. The *complete answer* to a query is the answer that could be computed if we could retrieve all the tuples from the relations in the query as if with no access limitations. In the following, we indicate with  $\text{ans}(q, \mathcal{R}, D)$  the set of tuples in the maximal answer to a query  $q$  over a schema  $\mathcal{R}$  with access limitations on a database  $D$ . The maximal answer can be computed by means of a Datalog program, as described in [14].

A relation  $r$  in a relation schema  $\mathcal{R}$  is *relevant* for a query  $q$  if there are two database instances  $D$  and  $D'$  over  $\mathcal{R}$  for which  $\text{ans}(q, \mathcal{R}, D) \neq \text{ans}(q, \mathcal{R}, D')$  and such that  $r^D \neq r^{D'}$  and  $s^D = s^{D'}$  for every relation  $s \neq r$  in  $\mathcal{R}$ .

We show that relevance is undecidable in the case of Datalog queries, since, if we were able to decide relevance of a relation, we could also decide containment between Datalog queries, which is known to be undecidable.

**THEOREM 1.** *Testing relevance for Datalog queries is undecidable.*

**PROOF.** Let  $\Pi$  be a Datalog program over a schema  $\mathcal{R}$ , without access limitations, defining two arbitrary predicates  $p$  and  $q$ . Let  $e$  be an extensional predicate not occurring in  $\Pi$  and  $i$  a new intensional predicate defined by the rules  $i(\vec{X}) \leftarrow e, p(\vec{X})$ , and  $i(\vec{X}) \leftarrow q(\vec{X})$ . If we could establish whether  $e$  is relevant to answer the Datalog query  $\text{Ans}(\vec{X}) \leftarrow i(\vec{X})$  then we could also decide containment between  $p$  and  $q$ , which is absurd (and *a fortiori* absurd for a schema that can have access limitations). More precisely,  $q$  contains  $p$  iff  $e$  is not relevant, i.e.:

- (1) If  $q$  contains  $p$ , then  $e$  is not relevant.
- (2) If  $q$  does not contain  $p$ , then  $e$  is relevant.

To see that (1) holds, it suffices to observe that  $p$  cannot contribute any tuple to  $i$  that is not already contributed by  $q$ , and thus  $e$  need not be accessed.

To see that (2) holds, consider a database  $D$  in which  $p(\vec{c})$  holds but  $q(\vec{c})$  does not hold, for some constants  $\vec{c}$ . Such a database must exist, as  $p$  is not contained in  $q$ . Since  $p$  and  $q$  are independent of  $e$ , their containment is also independent of  $e$ , so  $e$  may either hold or not hold in  $D$ . If  $e$  holds in  $D$ , then  $\vec{c}$  is in the answer; if  $e$  does not hold in  $D$ , then  $\vec{c}$  is not in the answer. Therefore  $e$  is relevant, since it changes the maximal answer to the query.  $\square$

## 2. REFERENCES

- [1] M. Benedikt, G. Gottlob, and P. Senellart. Determining relevance of accesses at runtime. In M. Lenzerini and T. Schwentick, editors, *PODS*, pages 211–222. ACM, 2011.
- [2] A. Cali, D. Calvanese, and D. Martinenghi. Optimization of query plans in the presence of access limitations. In M. Arenas and J. Hidders, editors, *Emerging Research Opportunities in Web Data Management (EROW 2007)*, pages 33–47. Informal proceedings, 2007.
- [3] A. Cali, D. Calvanese, and D. Martinenghi. Dynamic Query Optimization under Access Limitations and Dependencies. *Journal of Universal Computer Science*, 15(21):33–62, 2009.
- [4] A. Cali and D. Martinenghi. Conjunctive Query Containment under Access Limitations. In *ER 2008*, pages 326–340, 2008.
- [5] A. Cali and D. Martinenghi. Querying Data under Access Limitations. In *ICDE 2008*, pages 50–59, 2008.
- [6] A. Cali and D. Martinenghi. Optimizing Query Processing for the Hidden Web (Tutorial). In *Advances in Web Technologies and Applications, Proceedings of the 12th Asia-Pacific Web Conference, APWeb 2010, Busan, Korea, 6-8 April 2010*, page 397, 2010.
- [7] A. Cali and D. Martinenghi. Querying the deep web (tutorial). In *EDBT 2010, 13th International Conference on Extending Database Technology, Lausanne, Switzerland, March 22-26, 2010, Proceedings*, pages 724–727, 2010.
- [8] A. Cali, D. Martinenghi, and D. Carbotta. Query optimisation for web data sources: minimisation of the number of accesses (Extended Abstract). In *SEBD 2007*, pages 316–323, 2007.
- [9] H. Christiansen and D. Martinenghi. Symbolic Constraints for Meta-Logic Programming. *Applied Artificial Intelligence*, 14(4):345–367, 2000.
- [10] H. Christiansen and D. Martinenghi. Simplification of database integrity constraints revisited: A transformational approach. In *Logic Based Program Synthesis and Transformation, 13th International Symposium LOPSTR 2003, Uppsala, Sweden, August 25-27, 2003, Revised Selected Papers*, volume 3018 of *Lecture Notes in Computer Science*, pages 178–197. Springer, 2004.
- [11] H. Decker and D. Martinenghi. Avenues to flexible data integrity checking. In *Proceedings of the International Workshop on Flexible Database and Information System Technology (FlexDBIST-06)* 6 September 2006, Krakow, Poland, pages 425–429. IEEE Computer Society, 2006.
- [12] H. Decker and D. Martinenghi. Getting rid of straitjackets for flexible integrity checking. In *Proceedings of the 2nd International Workshop on Flexible Database and Information System Technology (FlexDBIST-07)*, pages 360–364, 2007.
- [13] A. Deutsch, B. Ludäscher, and A. Nash. Rewriting queries using views with access patterns under integrity constraints. *Theor. Comp. Sci.*, 371(3):200–226, 2007.
- [14] O. M. Duschka, M. R. Genesereth, and A. Y. Levy. Recursive query plans for data integration. *J. Log. Program.*, 43(1):49–73, 2000.
- [15] O. M. Duschka and A. Y. Levy. Recursive plans for information gathering. In *Proc. of IJCAI’97*, pages 778–784, 1997.
- [16] D. Florescu, A. Y. Levy, I. Manolescu, and D. Suciu. Query optimization in the presence of limited access patterns. In *Proc. of ACM SIGMOD*, pages 311–322, 1999.
- [17] A. Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294, 2001.
- [18] A. Y. Levy. Answering queries using views: A survey. Technical report, University of Washington, 1999.
- [19] C. Li. Computing complete answers to queries in the presence of limited access patterns. *VLDB Journal*, 12(3):211–227, 2003.
- [20] C. Li and E. Chang. Query planning with limited source capabilities. In *Proc. of ICDE 2000*, pages 401–412, 2000.
- [21] C. Li and E. Chang. Answering queries with useful bindings. *ACM Trans. on Database Systems*, 26(3):313–343, 2001.
- [22] C. Li and E. Chang. On answering queries in the presence of limited access patterns. In *Proc. of ICDT 2001*, pages 219–233, 2001.
- [23] D. Martinenghi. Simplification of integrity constraints with aggregates and arithmetic built-ins. In *Flexible Query Answering Systems, 6th International Conference, FQAS 2004, Lyon, France, June 24-26, 2004, Proceedings*, volume 3055 of *Lecture Notes in Computer Science*, pages 348–361. Springer, 2004.
- [24] D. Martinenghi. Access pattern. In H. C. van Tilborg and S. Jajodiathe, editors, *Encyclopedia of Cryptography and Security (Second Edition)*, pages A17–A20. Springer, 2011.
- [25] T. D. Millstein, A. Y. Halevy, and M. Friedman. Query containment for data integration systems. *J. of Computer and System Sciences*, 66(1):20–39, 2003.
- [26] T. D. Millstein, A. Y. Levy, and M. Friedman. Query containment for data integration systems. In *Proc. of PODS 2000*, pages 67–75, 2000.
- [27] A. Nash and B. Ludäscher. Processing first-order queries under limited access patterns. In *Proc. of PODS 2004*, pages 307–318, 2004.
- [28] G. Yang, M. Kifer, and V. K. Chaudhri. Efficiently ordering subgoals with access constraints. In *Proc. of PODS 2006*, pages 22–22, 2006.